**表现层设计文档**

架构

2013-11-22

**修改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | Jira号 | 修改日期 | 修改人员 | 修改简述 |
| 1 | MUSHROOM-25 | 2013-11-22 | 倪俊佳 | 创建文档，总述 |
| 2 | MUSHROOM-25 | 2013-12-16 | 倪俊佳 | RESTful API |

# 展示层总述

展示层目前主要指浏览器和WEB服务器这一层面。后期可指运行于各种操作系统之上的APP及服务器。

WEB服务器主要的作用是充当计算层和数据交互层以及系统的部分权限控制器。接受用户（浏览器）的数据请求。目前主要是两类数据请求，一是数据查看或者下载请求，告知研究人员目前各个设备以及传感器采集参数情况。二是设备控制请求，这个请求数据将由WEB服务器通过TCP通信转交到中间件层，并等待处理结果，以此来告知用户此次执行是否成功。部分权限控制主要是指将使用此类系统的人员进行分组，分配权限，在最大程度上保证系统安全合理运行。

浏览器主要从服务器拿取数据，之后绘制相应的图表。另外提供数据下载接口。在前期还承担用户输入指令的作用，比如输入设备计划运行时间表，以及直接控制设备的指令。

# 展示层界面设计

## 呈现数据

这段主要介绍数据怎样呈现。

对于大数据如何呈现：

图片or直接传输or压缩

## 数据处理

这段主要描写在线处理数据。

## 下载数据

这段主要介绍数据下载支持的格式

## 控制设备

这段主要介绍如何具体控制设备

# 浏览器和WEB服务器之间的信息交互

## REST API

在 REST 中，开发人员显式地使用 HTTP 方法，对系统资源进行创建、读取、更新和删除的操作：

* 使用 POST 方法在服务器上创建资源
* 使用 GET 方法从服务器检索某个资源或者资源集合
* 使用 PUT 方法对服务器的现有资源进行更新
* 使用 DELETE 方法删除服务器的某个资源

更为标准的内容协商方式是使用 HTTP 头。我们通常使用 Accept 来设置我们接受的返回结果的内容格式，用 Accept-Charset 来设置字符集，用 Accept-Encoding 来设置数据传输格式，用 Accept-Language 来设置语言。

## 数据传输格式

默认出错信息全部会提交到信息提示层，数据传输格式如[Alert 信息提示框](#_Alert_信息提示框)中所示。另外对于没有具体返回数据时的请求也会在信息提示层中显示，所以下面不再累述。

### Slider 侧滑导航

|  |
| --- |
| **// user login**  **POST /login**  **{**  **"userName": [userName]**  **"password": [password]**  **}**  **Response:**  **{**  **"token": [token] // 授权Session Token**  **"gravatar": [uri]**  **"userId": [userId]**  **"userName": [userName]**  **"groupId": [groupId]**  **"groupName": [groupName]**  **"menu": {**  **"mushroom": [uri]**  **"profile": [uri]**  **}**  **"extra": {**  **"version": "v0.0.1"**  **}**  **}** |

### Alert 信息提示框

|  |
| --- |
| // long polling or polling …  // maybe just show not communicate with serve  // Status Code Definitions  // 200 已创建，请求成功且服务器已创建了新的资源。  // 201 是否只显示处于警告状态的应用实例  // 301 重定向 , 请求的网页已被永久移动到新位置。服务器返回此响应时，会自动将请求者转到新位置。  // 302 重定向 , 请求的网页临时移动到新位置，但求者应继续使用原有位置来进行以后的请求。302 会自动将请求者转到不同的临时位置。  // 304 未修改，自从上次请求后，请求的网页未被修改过。服务器返回此响应时，不会返回网页内容。  // 400 错误请求 , 服务器不理解请求的语法。  // 401 未授权 , 请求要求进行身份验证。  // 403 已禁止 , 服务器拒绝请求。  // 404 未找到 , 服务器找不到请求的网页。  // 405 方法禁用 , 禁用请求中所指定的方法。  // 406 不接受 , 无法使用请求的内容特性来响应请求的网页。  // 408 请求超时 , 服务器等候请求时超时。  // 410 已删除 , 如果请求的资源已被永久删除，那么，服务器会返回此响应。  // 412 未满足前提条件 , 服务器未满足请求者在请求中设置的其中一个前提条件。  // 415 不支持的媒体类型 , 请求的格式不受请求页面的支持。  // 500 内部服务器错误。  **//GET /alert**  **//Response:**  **{**  **"code": [code]**  **"definitions": [definitions]**  **}** |

### Room Thumbnail 房间缩略卡片（即房间缩略信息）

|  |
| --- |
| // ============================================  // get room’s thumbnail  // return an array, erery item is an object like this below  **GET /rooms**  **Response:**  **[**  **{**  **"roomId": [roomId]**  **"roomName": [roomName]**  **"plantId": [plantId]**  **"plantName": [plantName]**  **"time": [time]**  **"sensors": {**  **"temperature": [value]**  **"co2":[value]**  **"humidity": [value]**  **"specturm": [value]**  **}**  **// here uri is meaning for image or icon**  **"menu": {**  **"data": [uri]**  **"rules": [uri]**  **"control": [uri]**  **}**  **}**  **……**  **]**  // ============================================  // get room’s thumbnail by room id  // return an array, erery item is an object like this below  **GET /room/[id]**  **Response:**  **{**  **"roomId": [roomId]**  **"roomName": [roomName]**  **"plantId": [plantId]**  **"plantName": [plantName]**  **"time": [time]**  **"sensors": {**  **"temperature": [value]**  **"co2":[value]**  **"humidity": [value]**  **"specturm": [value]**  **}**  **// here uri is meaning for image or icon**  **"menu": {**  **"data": [uri]**  **"rules": [uri]**  **"control": [uri]**  **}**  **}**  // ============================================  // update room’s name  **PUT /room/[id]/name**  **{**  **"roomId": [roomId]**  **"roomName": [roomName]**  **}**  **Response as alert**  // ============================================  // update plant’s name  **PUT /plant/[id]/name**  **{**  **"plantId": [plantId]**  **"plantName": [plantName]**  **}**  **Response as alert** |

### Search 搜索功能

|  |
| --- |
| // ============================================  // get data by room id  **GET /search**  **{**  **"roomId": [roomId]**  **"sensorId": [sensorId]**  **"startTime": [startTime]**  **"endTime": [endTime]**  **}**  **Response:**  **{**  **"roomId": [roomId]**  **"roomName": [roomName]**  **"sensorId": [sensorId]**  **"sensorName": [sensorName]**  **"value": [**  **{**  **"time": "value"**  **}**  **...**  **]**  **}** |

### Room Environment Data 房间环境数据（以单个图表为例）

|  |
| --- |
| // get all sensor information in a room  // return an array, erery item is an object like this below  **GET /room/[id]/environment**  **Response:**  **[**  **{**  **"sensorId": [sensorId]**  **"sensorName": [sensorName]**  **"value": [**  **{**  **[time]: [value]**  **...**  **}**  **]**  **}**  **...**  **]** |

### Plant Rules 养殖模式 （设置模式和查看模式）

|  |
| --- |
| // ============================================  // get all plant rules thumbnails (just name)  **GET /rules**  **Response:**  **[**  **{**  **"ruleId": [ruleId]**  **"ruleName": [ruleName]**  **}**  **...**  **]**  // ============================================  // get the detail by ruleId  **GET /rule/[ruleId]**  **Response:**  **{**  **"ruleId": [ruleId]**  **"ruleName": [ruleName]**  **"rules": [**  **{**  **"time": [time]**  **"capture": [CaptureObject]**  **"range": [min, max]**  **}**  **...**  **]**  **}**  // ============================================  // set a rule  **POST /rule**  **{**  **"ruleName": [ruleName]**  **"rules": [**  **{**  **"time": [time]**  **"capture": [CaptureObject]**  **"range": [min, max]**  **}**  **...**  **]**  **}**  **Response as alert**  // ============================================  // update a rule’s name  **PUT /rule/[id]/name**  **{**  **"ruleId": [ruleId]**  **"ruleName": [ruleName]**  **}**  **Response as alert**  // ============================================  // delete a rule’s name  **DELETE /rule/[id]**  **{**  **"ruleId": [ruleId]**  **}**  **Response as alert** |

### Room Environment Controller 房间控制器

|  |
| --- |
| // ============================================  // get room’s controller status  **GET /room/[id]/controllers**  **Response:**  **[**  **{**  **"roomId": [roomId]**  **"controllerId": [controllerId]**  **"controllerType": [controllerType]**  **"state": [state]**  **}**  **…**  **]**  // ============================================  // get room’s controller status by controller id  **GET /room/[id]/controller/[id]**  **Response:**  **{**  **"roomId": [roomId]**  **"controllerId": [controllerId]**  **"controllerType": [controllerType]**  **"state": [state]**  **}**  // ============================================  // set room’s controller status by controller id  **PUT /room/[id]/controller/[id]**  **{**  **"roomId": [roomId]**  **"controllerId": [controllerId]**  **"action": [action]**  **}**  **Response as alert** |

### System Log View 系统日志查看

|  |
| --- |
| // ============================================  //get system log  **GET /system/log/error**  **GET /system/log/working**  **GET /system/log/communication**  **GET /system/log/debug**  **Response:**  **[**  **{**  **"time": [time]**  **"type": [type]**  **"message": [message]**  **"fun": [fun]**  **}**  **...**  **]** |

### User’s Basic Information 用户基本信息查看

|  |
| --- |
| // ============================================  //get all user’s basic information  GET /user/[id]  **Response:**  **{**  **"userId": [userId]**  **"userName": [userName]**  **"gravatar": [uri]**  **"email": [email]**  **"phone": [phone]**  **}**  **// ============================================**  **//reset user’s password**  **PUT /user/[id]/password**  **Response:**  **{**  **"userId": [userId]**  **"password1": [password1]**  **"password2": [password2]**  **}** |

### Administrator’s function 超级用户的功能

|  |
| --- |
| // ============================================  //add a user  POST /user  {  "userName": [userName]  "password": [password]  "email": [email]  "phone": [phone]  "group": [group]  }  Response as alert  // ============================================  //get all group names  GET /groups  [  {  "groupId": [groupId]  "groupName": [groupName]  }  ...  ]  Response as alert  // ============================================  //change a user group  PUT /user/[id]/group  {  "userId": [userId]  "group": [group]  }  Response as alert  // ============================================  //get all users names  GET /users  Response:  [  {  "userId": [userId]  "userName": [userName]  "email": [email]  "phone": [phone]  "group": [group]  }  ...  ] |

# 权限分配

## 拥有的权限

1. 设置设备计划任务
2. 设备的直接控制
3. 数据查看功能（不知后期是否需要细分）
4. 数据下载功能
5. 系统运行状况查看

## 系统中三类组

1. 系统超级管理员组（admins），拥有所有权限；
2. 研究人员组（researchers），主要是指蘑菇房的研究人员，拥有查看所有数据和控制蘑菇房设备的权限；
3. 系统游客组（guests），拥有查看数据的功能；
4. 系统维护人员组（repairs），查看系统运行状况。

# 系统安全性

## CSRF

CSRF（Cross-site request forgery跨站请求伪造，也被称为“one click attack”或者session riding，通常缩写为CSRF或者XSRF，是一种对网站的恶意利用。尽管听起来像跨站脚本（XSS），但它与XSS非常不同，并且攻击方式几乎相左。XSS利用站点内的信任用户，而CSRF则通过伪装来自受信任用户的请求来利用受信任的网站。与XSS攻击相比，CSRF攻击往往不大流行（因此对其进行防范的资源也相当稀少）和难以防范，所以被认为比XSS更具危险性。

## XSS

XSS又叫CSS (Cross Site Script) ，跨站脚本攻击。它指的是恶意攻击者往Web页面里插入恶意html代码，当用户浏览该页之时，嵌入其中Web里面的html代码会被执行，从而达到恶意用户的特殊目的。

## 其他安全问题

是否使用HTTPs解决一些问题？

# 参考

1. [虚拟研讨会：如何设计好的RESTful API？](http://www.infoq.com/cn/articles/how-to-design-a-good-restful-api)
2. [最佳实践：更好的设计你的 REST API](http://www.ibm.com/developerworks/cn/web/1103_chenyan_restapi/)